**Amendment to the Claims**

1. (CURRENTLY AMENDED) A method of communicating between a first process on a first computer and a destination on a second computer, the method comprising:

receiving an environment variable from the first process wherein the environment variable comprises destination information indicative of the destination;

determining a transport protocol at run-time for a message object based on the destination information by selecting a one of a process to process transport protocol, a process to pathway transport protocol, or a process to socket transport protocol, wherein the second computer comprises a Tandem parallel processor computer system and wherein the destination operates within a Guardian operating environment;

generating the message object based on the destination information;

receiving a pointer to a message to be transmitted from the first process to the destination;

transmitting the message from the first process to the destination using the message object and the determined transport protocol;

generating a tag associated with the message object transmitted from the first process to the destination; and

storing the tag in an outbound message vector.

2. (ORIGINAL) The method of claim 1 wherein the destination comprises a second process on the second computer.

3. (ORIGINAL) The method of claim 2 wherein the destination information comprises a name of the second process on the second computer.

4. (ORIGINAL) The method of claim 2 wherein the transport protocol comprises process to process.

5. (ORIGINAL) The method of claim 1 wherein the destination comprises a pathway to the second computer.

2

6. (ORIGINAL) The method of claim 5 wherein the destination information comprises a name of the pathway to the second computer.

7. (ORIGINAL) The method of claim 5 wherein the transport protocol comprises process to pathway.

8. (ORIGINAL) The method of claim 1 wherein the destination comprises a socket on the second computer.

9. (ORIGINAL) The method of claim 8 wherein the destination information is an address of the second computer.

10. (ORIGINAL) The method of claim 9 wherein the address comprises an Internet Protocol address.

11. (ORIGINAL) The method of claim 9 wherein the address comprises a port number of the socket.

12. (ORIGINAL) The method of claim 8 wherein the transport protocol comprises process to socket.

13. PREVIOUSLY CANCELED.

14. PREVIOUSLY CANCELED.

15. PREVIOUSLY CANCELED.

16. PREVIOUSLY CANCELED.

17. PREVIOUSLY CANCELED.

3

18. (ORIGINAL) The method of claim 1 further comprising checking for errors.

19. (ORIGINAL) The method of claim 18 further comprising calling an error service in response to the error.

20. (ORIGINAL) The method of claim 1 further comprising generating a unique identifier for the message.

21. (ORIGINAL) The method of claim 1 further comprising receiving a reply from the second computer.

22. (ORIGINAL) The method of claim 1 further comprising registering a service.

23. (ORIGINAL) The method of claim 1 further comprising receiving the message ~~information~~ from the first process into the destination.

24. (CURRENTLY AMENDED) A software product for communicating between a first process on a first computer and a destination on a second computer, comprising:

communication software operational when executed by a processor to direct the processor to receive an environment variable from the first process wherein the environment variable comprises destination information indicative of the destination, at run-time determine a transport protocol for a message object based on the destination information <u>by selecting a one of a process to process transport protocol, a process to pathway transport protocol, or a process to socket transport protocol</u> wherein the second computer comprises a Tandem parallel processor computer system and wherein the destination operates within a Guardian operating environment, generate the message object based on the destination information, receive a pointer to a message to be transmitted from the first process to the destination, transmit the message from the first process to the destination using the message object and the determined transport protocol,

4

generate a tag associated with the message object transmitted from the first process to the destination, and store the tag in an outbound message vector; and

a software storage medium operational to store the communication software.

25. (ORIGINAL) The software product of claim 24 wherein the destination comprises a second process on the second computer.

26. (ORIGINAL) The software product of claim 25 wherein the destination information comprises a name of the second process on the second computer.

27. (ORIGINAL) The software product of claim 25 wherein the transport protocol comprises process to process.

28. (ORIGINAL) The software product of claim 24 wherein the destination comprises a pathway to the second computer.

29. (ORIGINAL) The software product of claim 28 wherein the destination information comprises a name of the pathway to the second computer.

30. (ORIGINAL) The software product of claim 28 wherein the transport protocol comprises process to pathway.

31. (ORIGINAL) The software product of claim 24 wherein the destination comprises a socket on the second computer.

32. (ORIGINAL) The software product of claim 31 wherein the destination information is an address of the second computer.

33. (ORIGINAL) The software product of claim 32 wherein the address comprises an Internet Protocol address.

5

34. (ORIGINAL) The software product of claim 32 wherein the address comprises a port number of the socket.

35. (ORIGINAL) The software product of claim 31 wherein the transport protocol comprises process to socket.

36. PREVIOUSLY CANCELED.

37. PREVIOUSLY CANCELED.

38. PREVIOUSLY CANCELED.

39. PREVIOUSLY CANCELED.

40. PREVIOUSLY CANCELED.

41. (ORIGINAL) The software product of claim 24 wherein the communication software is operational when executed by the processor to direct the processor to check for errors.

42. (ORIGINAL) The software product of claim 41 wherein the communication software is operational when executed by the processor to call an error service in response to the error.

43. (ORIGINAL) The software product of claim 24 wherein the communication software is operational when executed by the processor to generate a unique identifier for the message.

44. (ORIGINAL) The software product of claim 24 wherein the communication software is operational when executed by the processor to receive a reply from the second computer.

45. (ORIGINAL) The software product of claim 24 wherein the communication software is operational when executed by the processor to register a service.

6